CMIF
Center for Multisource Information Fusion

An Algorithm-Level Test Bed for Level-One Data Fusion Research (CASE_ATTI)

Fu-Mao (Eric) Chou

Report No. CMIF 2-01

March 2001

C{U}B

CALSPAN–UNIVERSITY AT BUFFALO RESEARCH CENTER. INC.

University at Buffalo
The State University of New York

20010625 171

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-01-

0387

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 2001	3. REPORT TYPE AND DATES COVERED Final Report (1/1/99 – 12/31/00)

4. TITLE AND SUBTITLE An Algorithm-Level Test Bed for Level-one Data Fusion Research (CASE_ATTI)	6. FUNDING NUMBERS G-F49620-99-1-0090
5. AUTHORS Fu-Mao (Eric) Chou	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Calspan-UB Research Center, Inc. P.O. Box 400 4455 Genesee Street Buffalo, New York 14225	8. PERFORMING ORGANIZATION REPORT NUMBER CMIF 2-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research 801 N. Randolph Street, Room 732 Arlington, VA 22203-1977	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES None	AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFOSR) NOTICE OF TRANSMITTAL DTIC. THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLIC RELEASE LAW AFR 190-12. DISTRIBUTION IS UNLIMITED.
12a. DISTRIBUTION/AVAILABILITY STATEMENT Publicly releasable; unlimited distribution	

13. ABSTRACT *(Maximum 200 words)*

This report summarizes part of the research conducted at the Center for Multisource Information Fusion (CMIF) at the State University of New York at Buffalo (SUNY at Buffalo) during the second year of a two-year Air Force Office of Scientific Research (AFOSR)-funded research grant. The overarching research objective of this grant is to provide understanding about the nature of multi-platform and distributed data fusion and the influence that such methods might have on flight-testing of future multi-platform systems at major range facilities such as, in particular, Edwards Air Force Base (the Air Force Flight Test Center, AFFTC), and also with a special focus on Electronic Warfare (EW) aspects and impacts. This particular report describes a simulation-based research tool called "CASE-ATTI" (Concept Analysis and Simulation Environment for Automatic Target Tracking and Identification) that was used to conduct various other research projects within the overarching grant effort. This tool was graciously provided to CMIF by the Canadian Department of National Defense and the Defense Research Establishment, Valcartier (DREV, Quebec, Canada) in particular, for which we are very grateful. This tool is a state-of-the-art Level 1 data fusion research tool, focused on multisensor, fusion-based techniques for tracking and identification of single objects. It is typical of the type of tools that will be necessary at AFFTC for testing and evaluation of future data fusion-capable flight platforms. This report describes this advanced tool and an example of its application and use in a research task being conducted at CMIF.

14. SUBJECT TERMS Distributed Data Fusion, Flight Testing, Flight Test Planning, Flight Test Resources, Data Fusion Testbeds			15. NUMBER OF PAGES 36
			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT U	18. SECURITY CLASSIFICATION OF THIS PAGE U	19. SECURITY CLASSIFICATION OF ABSTRACT U	20. LIMITATION OF ABSTRACT UL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Background

Level-One Data Fusion (L1DF) is about tactical picture compilation. According to the literature (e.g. Ref 1), a L1DF system processes the information and data reported by multiple dissimilar sources in order to correctly and quickly derive the best estimates of the kinematic properties for each perceived entity in the environment, and develop inferences as to the identity and key attributes of these entities. L1DF is concerned solely with individual entities (aircraft, missiles, etc.) considered in isolation (i.e., teams, groups or formations of entities are not considered at this fusion level). The reality of this focus on individual entities in any specific, real-world case is of course limited by sensor resolution characteristics.

1.2 Motivation and Objectives

From a research and development (R&D) perspective, the developers of L1DF systems need capabilities that allow them to quantitatively assess if the algorithms and techniques of a proposed L1DF system are suitably working. A set of tools is thus required to support the designer/developer/user/operator in his quantitative assessment of the performance of these algorithms and techniques. In that respect, a highly modular, structured, and flexible test bed, called CASE_ATTI (Concept Analysis and Simulation Environment for Automatic Target Tracking and Identification, see Ref 2), has been developed at the Defense Research Establishment at Valcartier, Quebec, Canada (DREV), and has been installed at the Center for Multisource Information Fusion (CMIF) at the State University of New York at Buffalo, in part to support research on this AFOSR/AFFTC research program. Stimulation is not the only way to evaluate target-tracking systems of course; analytical methods, to include covariance analysis, can also be used (e.g. Ref 3, 4). However, to be correctly applied, the analytical methods require various assumptions to be valid. This is in large part the result of the fact that there is no closed-form mathematics that clearly links the various sub-processes (and algorithms) associated with target tracking operation. Stimulation-based performance analysis of course also has its own cost, in the sense of simulator development, debugging, and

possible computational complexity in application to difficult scenarios. However, with the generosity of DREV, CMIF was able to employ a relatively mature L1DF research tool for its basic research studies without incurring the development-based cost penalty.

This report to AFOSR/AFFTC reflects: (a) the characteristics of CASE-ATTI and our (CMIF) research to fully understand it, and (b) the ways in which we have applied CASE-ATTI for the particular purposes of developing a new Performance Evaluation (PE) approach to multi-target, multi-sensor tracking, a critical issue for future AFFTC flight-test experiments.

Figure 1. The role of CASE_ATTI

1.3   Features of CASE-ATTI

* **A highly modular, structured, and flexible test bed**

   In order to simply the construction of new L1DF systems, CASE_ATTI was developed as a modular structure. Every modifiable task is designed as an independent module. The major advantage is that changing the settings of one module won't affect the settings of other modules. This is achieved, however, at

2

the expense of additional processing complexity; this was considered the correct tradeoff for a research tool.

- **A proof-of-concept demonstrator to allow continuing exploration**

  As mentioned in the section on objectives, the main goal of CASE_ATTI is to test the performance (relationship of estimates to simulated truth states) of a given L1DF system. Employing this integrated software, users can create or modify their own new L1DF system by changing both the system architecture and the parameters, then use the capability of test and evaluation built in the CASE_ATTI PE module to assess the performance of the created L1DF system. A key aspect of our research on this project is to conduct research toward the development of an augmented methodology for the PE module.

- Provide the algorithm-level test and replacement capability required to study and compare the technical feasibility, applicability and performance of advanced, state-of-the-art L1DF techniques

1.4 CASE_ATTI High Level Structure

To be capable of being a L1DF test bed, CASE_ATTI contains three major modules; they are the Stimulation Module, L1DF Module, and Performance Evaluation Module, as shown in Figure 2.
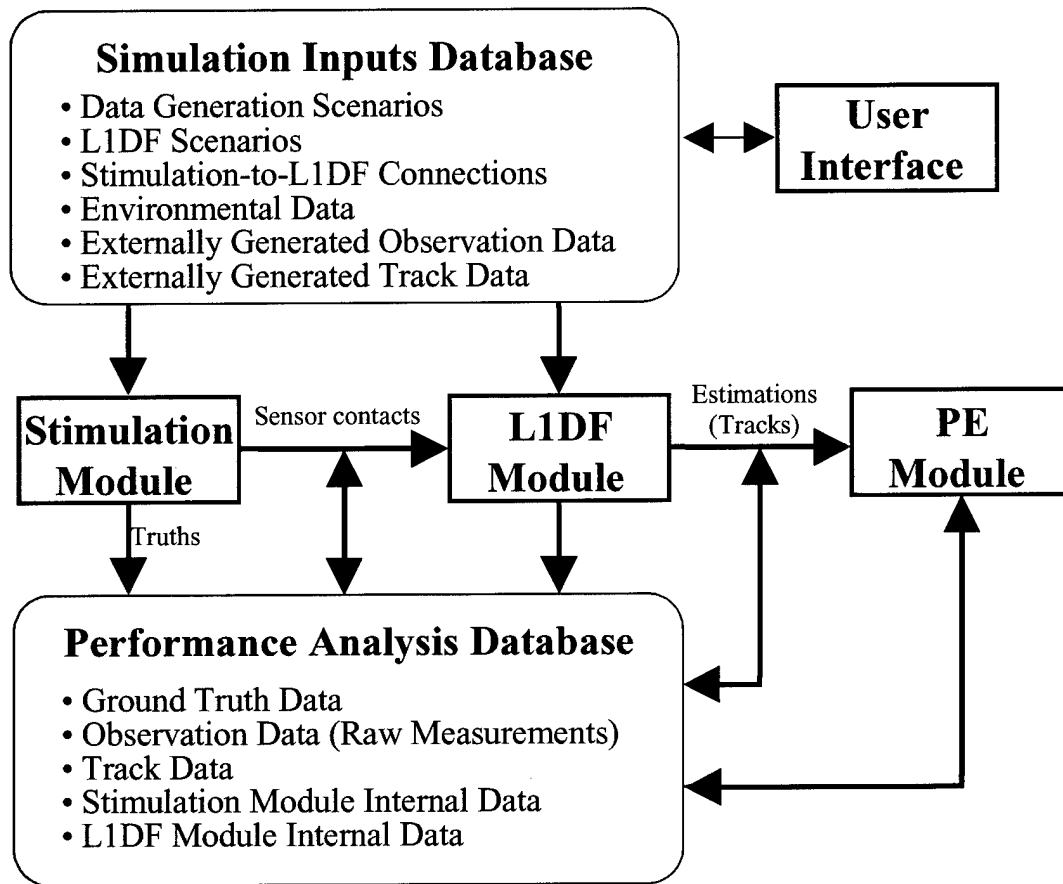
Figure 2. CASE_ATTI high-level structure

1.4.1 Stimulation Module

Since rigorously evaluating candidate L1DF algorithms using real data would not always be practical, CASE_ATTI has a high-fidelity stimulator that emulates the behavior of real targets, sensor systems and the meteorological environment, allowing the user to create and edit test scenarios with multiple ships/sensors/targets.

The stimulation module will generate two types of information, the truths and measurements/contacts. The measurements generated from the simulated sensors will then transfer into the next module (L1DF module) to generate the estimations. On the

other hand, the truth information will then be used to compare with the estimations in the last module (performance evaluation module).

1.4.2   1.4.2 L1DF Module

One of the main requirements of the CASE_ATTI test bed has also been to provide the algorithm-level test and replacement capability (required to study and compare the technical feasibility, applicability and performance of advanced, state-of-the-art L1DF techniques) where the user can switch between all available algorithms in the CASE_ATTI library without re-coding and/or re-compiling. The L1DF system module provides this capability and supports a wide variety of L1DF architecture types, varying from a simple single sensor tracker to an arbitrary complex hierarchical multiple sensor topology. Its design also has the capability of simulating a contact-level, track-level or hybrid fusion architecture as required. The data processing performed in this module is divided into a number of independent blocks such as target state estimation and kinematics behavior assessment, track and cluster management, data association, track fusion, target identity information fusion, sensor data alignment, and the internal system track database.

1.4.3   1.4.3 Performance Evaluation Module

A performance analysis database retains archives of all data manipulated. A performance evaluation module provides tools to assist the quantitative assessment of L1DF systems performance. A user interface module supports all interactions with the users/operators.

As alluded to above, the CASE_ATTI test bed is very modular to allow for maximum flexibility in the testing of various configurations and to allow for easy alteration or customizing in the future. In that respect, the Object-Oriented (OO) software design allows for the easy development and incorporation of new tracking and fusion algorithms, sensor models, and analysis tools.

CASE_ATTI runs on both UNIX and Windows platforms and the design also has the capabilities of utilizing multiple computers across a Local Area Network (LAN). Portability requirements have also driven the selection of the various software technologies used (e.g., C++, Oracle, OpenGL, Java, etc.).

Figure 3 shows the graphical user-interface of the CASE_ATTI high-level structure. The three rectangles from left to right are the stimulation module, L1DF module, and performance evaluation module respectively.
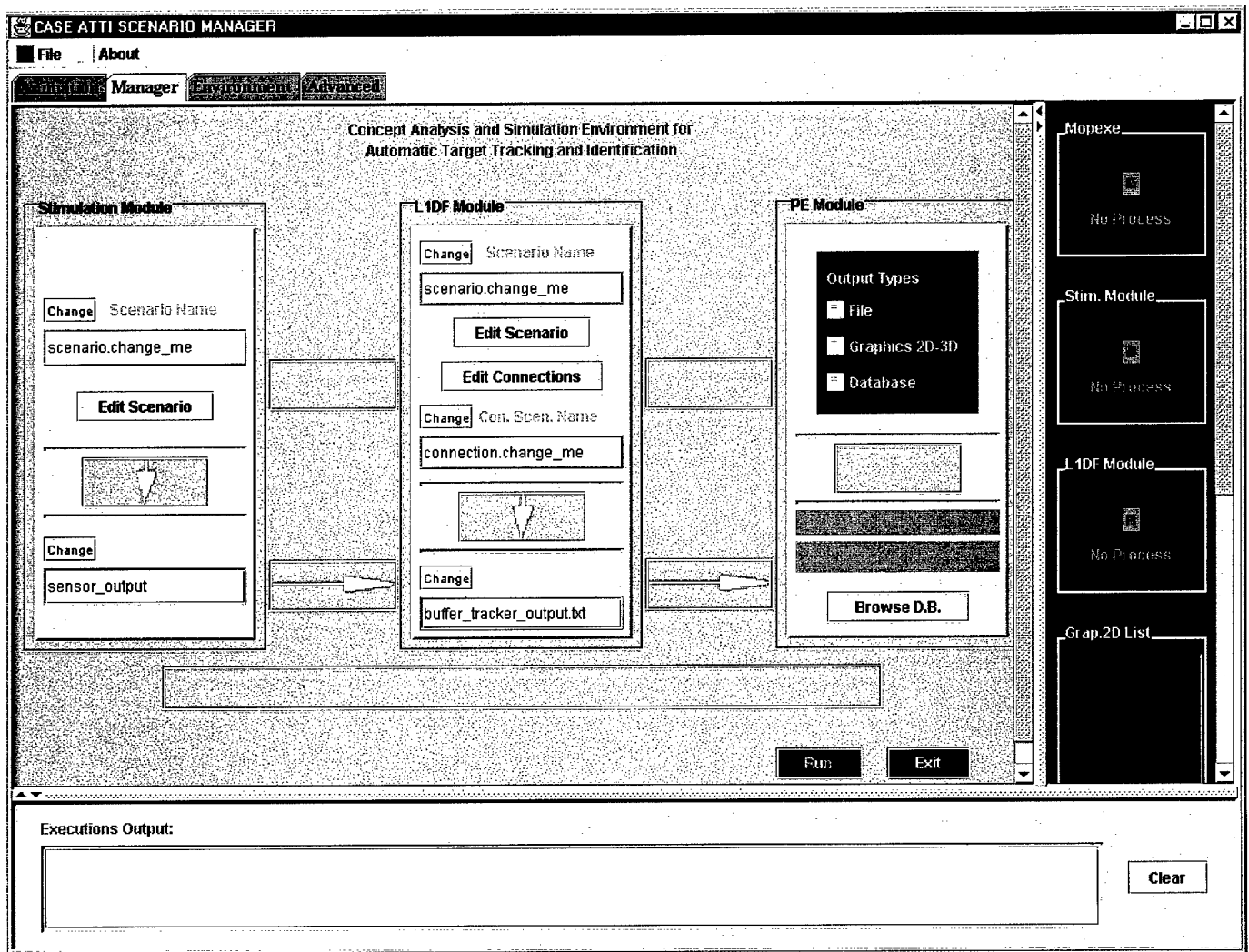

Figure 3. The user-interface of CASE_ATTI high-level structure

CHAPTER 2   DETAILED DESCRIPTION OF CASE-ATTI

2.1   Stimulation Module

It has been identified early in the L1DF project at DREV that rigorously evaluating candidate L1DF algorithms and techniques using real sensor and link data with meaningful measures of performance would not always be practical. For example, real data don't include ground truth information, thereby reducing the possibilities for comparing the *perceived* situation with the *real* situation in the environment. One also has to face the extremely limited availability of trial data to support algorithm analysis and development and L1DF system prototyping. Many research programs cannot afford to incur the additional costs of data collection for the purpose of demonstrating concepts with real data. Alternatives to this situation include artificially synthesizing appropriate data from trial data collected under non-standard conditions (not easy to do in a convincing manner), or to employ sufficiently high-fidelity simulators. This last option has been chosen for the R&D activities aiming at the exploration of the L1DF concepts at DREV since, most of the time, representative simulated data may be adequate to verify or validate L1DF concepts.

Hence, besides the possibility of using real data, a high-fidelity simulator that emulates the behavior of real targets, sensor systems and the meteorological environment relevant to the real world has been developed for CASE-ATTI. The Stimulation Module (SM) of CASE_ATTI (a.k.a. the "sensor" module) provides the realistic simulated sensor and link data required to stimulate the L1DF system module described below (a.k.a. the "tracking" module) so that the end user can assess the performance of the L1DF system under evaluation in representative conditions.

Figure 4 illustrates a typical test scenario in CASE_ATTI. The stimulation module allows the user to create and edit test scenarios with multiple ships/sensors/targets. The ships can be stationary or moving along user-predefined paths. One or several sensors can be assigned to each ship (currently, the stimulation module supports surveillance radar, IFF, ESM and IR sensor and link simulations). Targets are created with user-predefined 3D

trajectories and attributes. Figure 4 uses the terms "platforms" and "targets" to distinguish the difference here.
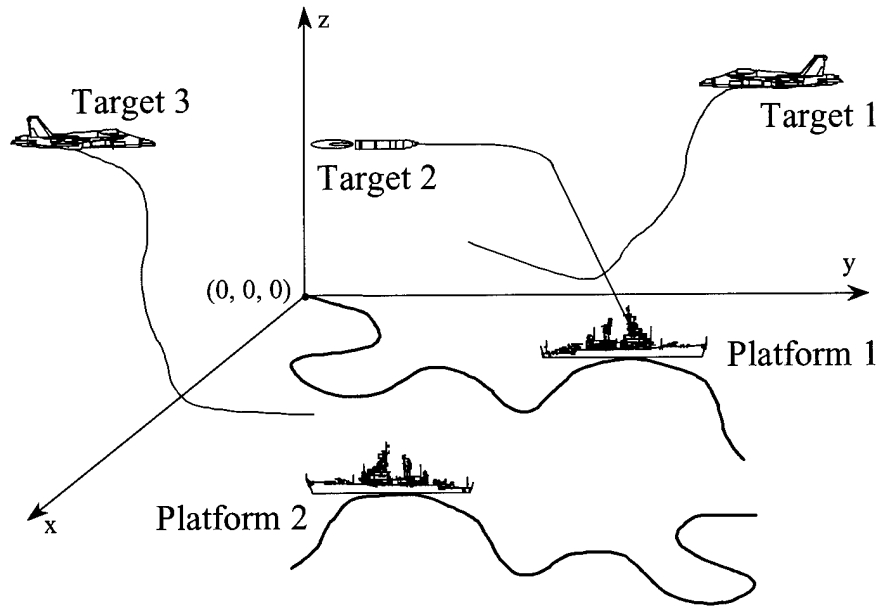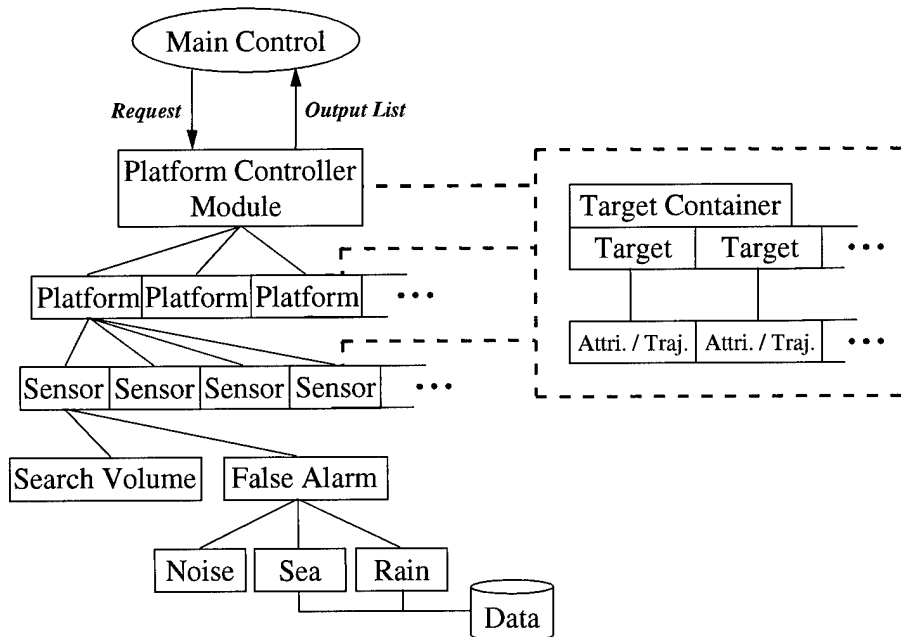

Figure 4. Typical test scenario

Figure 5. CASE_ATTI stimulation module

Ultimately, given such a test scenario, the stimulation module generates true target kinematics and measured target kinematics, which are then made available to the L1DF system module. At a very high level, the interactions of the main objects of stimulation module are illustrated in Figure 5; the functioning of the stimulation module will be elaborated on the following sections.

In the stimulation module, three types of elements have to be defined to simulate the real world environment; they are the platforms, sensors, and targets.

2.1.1 Platforms

The term platform can mean a variety of things; in CASE-ATTI it is defined as follows: "A platform is an object that holds one or multiple sensors. A platform may be an airplane, a ship or even a submarine. It can be stationary or it can be moving along a user-predefined path. The platform is the object from which we observe the environment. It is itself a target because it shares the characteristics (trajectory and target type) of targets."

The stimulation module is built on several levels of abstraction. The platform controller can be thought of as a hub that is responsible for collecting data from the various platforms, and for organizing this data for distribution either to the L1DF module or to the user interface. The responsibilities of this module are thus to create and initialize each platform in the scenario and to manage these platforms after each configuration. The latter involves updating each platform and requesting the correct platform for observation data. At this highest level of abstraction, the platform controller hides the details of sensor simulation from the programmer. The programmer knows there is an object that controls the platforms in the scenario and that this object will return observation data, but how these data are actually generated is unknown to his perspective.

All sensors are mounted on some type of platform that can be moving or be stationary. A platform can also be underwater, on the surface or airborne. Each platform must react to requests from the platform controller. Typical requests are to advance to the next event time and to obtain from the platform its location, the type of sensors assigned to it, and a list of observations. This is the second level of abstraction.

9

2.1.2 Sensors

"A platform perceives its environment using sensor objects. One or several sensors can be assigned to a platform". In the current version of CASE-ATTI, the stimulation module supports the following types of sensors:

- *Surveillance Radar*
- *IFF (Identification of Friend or Foe)*
- *ESM (Electronic Support Measure)*
- *IR sensor (Infra-Red)*

At the next level of abstraction, the sensor object represents the specific type of sensor that is being modeled. As shown in Figure 5, the sensor object collaborates with several other objects to create the appropriate observations. Collaboration occurs with the target container, false alarm and search volume objects in producing the required observations. The target container manages all targets in the simulated scenario and all requests for targets must pass through this class. The target object is an abstraction of a real-world object that can be viewed by a sensor.

Since a platform may contain several dissimilar sensors, a common interface is used for each sensor to increase reuse and extensibility. Each sensor (be it an infrared, a radar, a data file reader, etc.) can have a completely different representation beyond this interface and, therefore, previously derived sensor models are not altered by the addition of a new sensor model.

To simulate clutter and other corrupting effects in the real world, CASE_ATTI can simulate the false alarms when sensors are "detecting" the contacts from targets. The false alarms include noise clutter, weather clutter, and sea clutter.

2.1.3 Targets

A target is an object that could be detected by a sensor. Meteorological phenomena that could produce false alarms are not considered to be a target. We deliberately use the expression "could be detected" because the actual detection of a target is not always

10

guaranteed. It depends on the setting of the sensor, the sensor-target geometry, etc. Examples of targets include ships, airplanes, missiles, etc.

2.1.4 Modifiable Parameters for the Stimulation Module

Most of the modifiable parameters are listed in this section. By changing the settings of those parameters corresponding to platforms, sensors, and targets, users can create a unique test scenario that could be used to test a certain L1DF system.

For platforms and targets:

- Generic attributes and characteristics: (such as dimension and identity)
- Trajectory: (moving or stationary)

For sensors:

- Generic attributes and characteristics:

    - *Type of sensors*
    - *Probability of detection*

- Search volume:

    - *Maximum and Minimum range*
    - *Scan RPM*

- False alarms:

    - *Noise clutter*
    - *Weather clutter*
    - *Sea clutter*

2.1.5 User-interface of the Stimulation Module

Figure 6 shows the user-interface of the scenario editor in the stimulation module. The parameters listed above can be set or changed by calling out the required window.
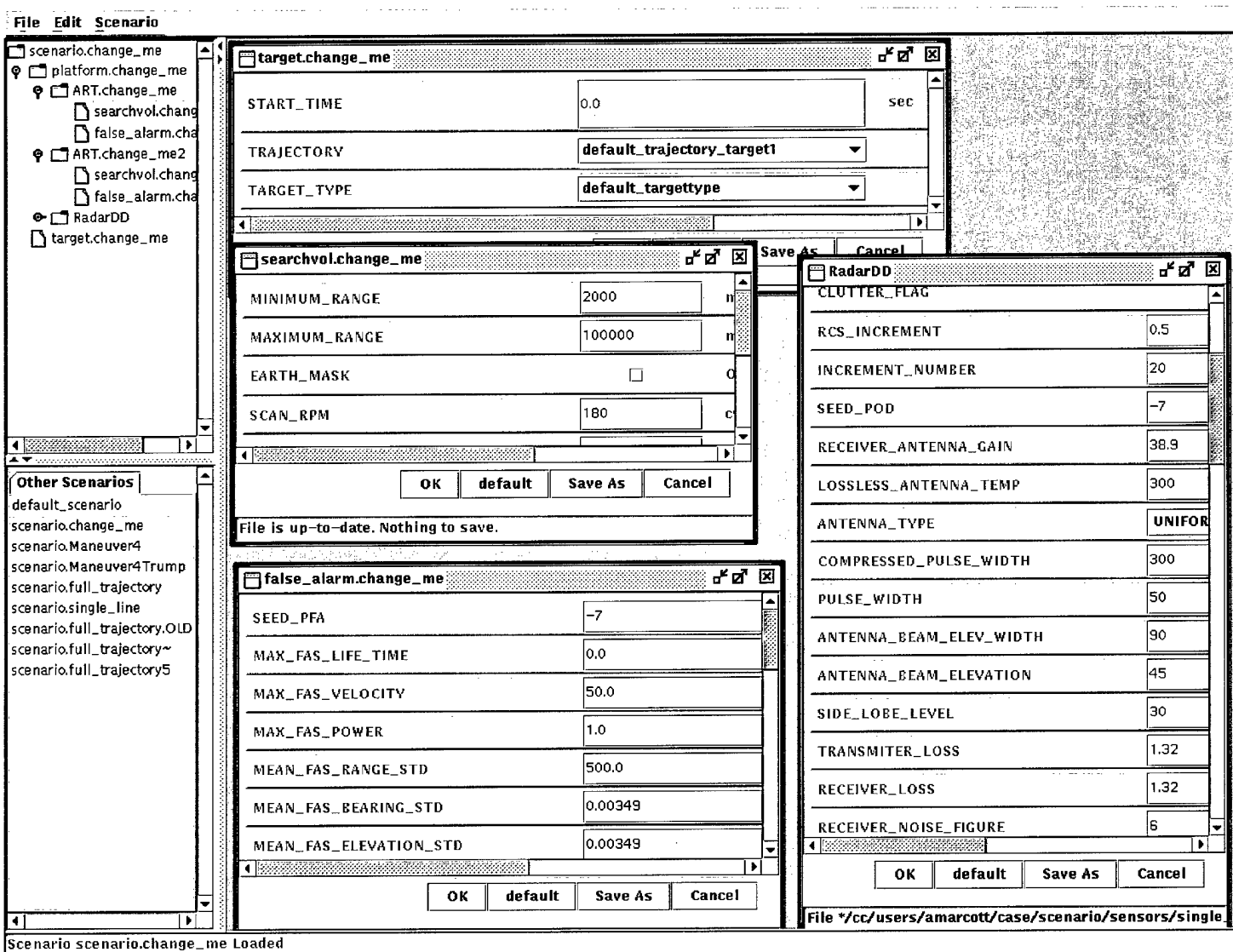
File Edit Scenario

scenario.change_me
platform.change_me
ART.change_me
searchvol.chang
false_alarm.cha
ART.change_me2
searchvol.chang
false_alarm.cha
RadarDD
target.change_me

target.change_me

START_TIME	0.0	sec
TRAJECTORY	default_trajectory_target1 ▼	
TARGET_TYPE	default_targettype ▼	

Save As   Cancel

searchvol.change_me

MINIMUM_RANGE	2000	n
MAXIMUM_RANGE	100000	n
EARTH_MASK	☐	c
SCAN_RPM	180	c

OK   default   Save As   Cancel

File is up-to-date. Nothing to save.

RadarDD

CLUTTER_FLAG	
RCS_INCREMENT	0.5
INCREMENT_NUMBER	20
SEED_POD	-7
RECEIVER_ANTENNA_GAIN	38.9
LOSSLESS_ANTENNA_TEMP	300
ANTENNA_TYPE	UNIFOR
COMPRESSED_PULSE_WIDTH	300
PULSE_WIDTH	50
ANTENNA_BEAM_ELEV_WIDTH	90
ANTENNA_BEAM_ELEVATION	45
SIDE_LOBE_LEVEL	30
TRANSMITER_LOSS	1.32
RECEIVER_LOSS	1.32
RECEIVER_NOISE_FIGURE	6

OK   default   Save As   Cancel

File */cc/users/amarcott/case/scenario/sensors/single

Other Scenarios

default_scenario
scenario.change_me
scenario.Maneuver4
scenario.Maneuver4Trump
scenario.full_trajectory
scenario.single_line
scenario.full_trajectory.OLD
scenario.full_trajectory~
scenario.full_trajectory5

false_alarm.change_me

SEED_PFA	-7
MAX_FAS_LIFE_TIME	0.0
MAX_FAS_VELOCITY	50.0
MAX_FAS_POWER	1.0
MEAN_FAS_RANGE_STD	500.0
MEAN_FAS_BEARING_STD	0.00349
MEAN_FAS_ELEVATION_STD	0.00349

OK   default   Save As   Cancel

Scenario scenario.change_me Loaded

Figure 6. The user-interface of CASE_ATTI stimulation module

2.1.6   Results of the Stimulation Module

To monitor the created scenario, CASE_ATTI provides a graphic module allowing users to see the results in a graphic form. Users can select the part of results they only want to see. Figure 7 shows representative graphic outputs from the stimulation module. There are four targets moving from the upper-right corner diagonally toward the lower-left, then maneuvering outward.

12

Figure 7(a) shows all the possible information that can be generated from stimulation module including "true-tracks", "contacts/measurements", and "false alarm" (if activated). Figure 7(b) shows only the "true-tracks" and "contacts/measurements". The "true-tracks" and "contacts/measurements" information are shown in Figure 7(c) and Figure 7(d) respectively.
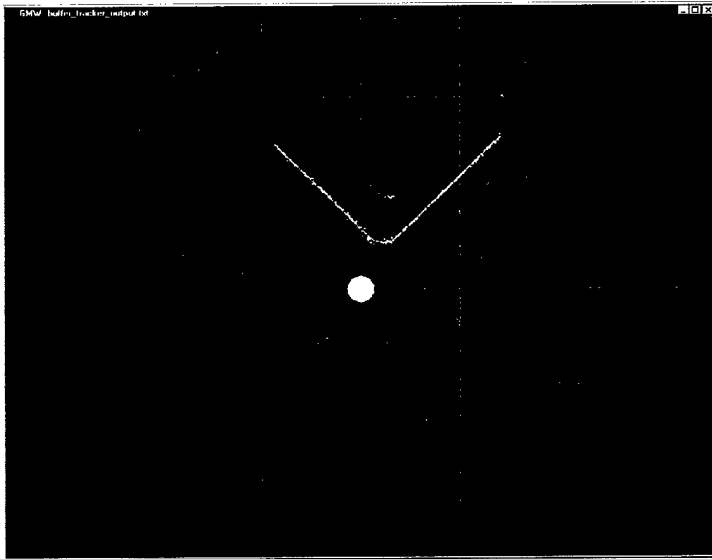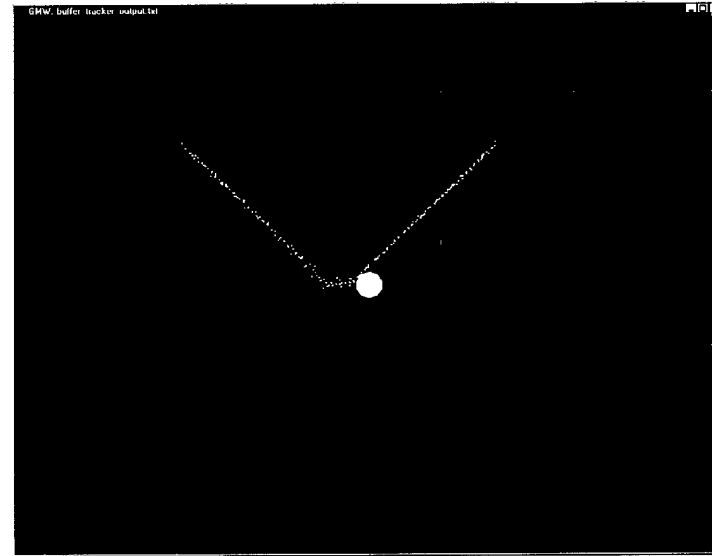

(a) True-tracks + contacts + false alarms

(b) True-tracks + contacts
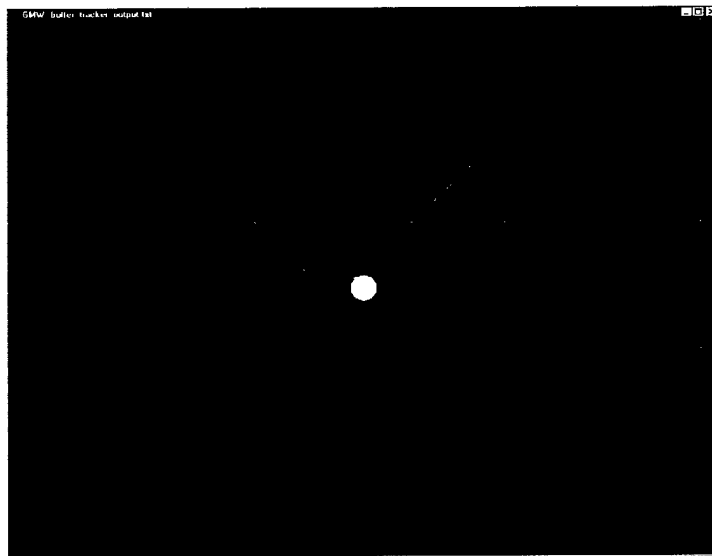
(c) True-tracks

(d) Contacts

Figure 7. Graphic viewer of CASE_ATTI (Stimulation Module)

2.2 L1DF Module

In addition to high-fidelity stimulation, one of the main requirements of the CASE_ATTI test bed has been to provide the algorithm-level test and replacement capability required to study and compare the technical feasibility, applicability and performance of advanced, state-of-the-art L1DF techniques. Many test beds developed for R&D in L1DF support parametric-level but not algorithm-level experimentation. That is, these test beds have typically been built for a given particular application wherein only the configuration parameters of the specific algorithm implemented can be altered to study attendant effects. However, these test beds usually don't allow the easy replacement of the complete algorithm itself, globally at once, as a single block, without major modifications. Hence, one of the main requirements of the L1DF system module of CASE_ATTI has thus been to provide this algorithm-level replacement capability where the user can switch between all available algorithms in the CASE_ATTI library without re-coding and/or re-compiling.

To ensure a proper development of a software model and to facilitate its maintenance, it is very important to identify the main objectives to be pursued early during the analysis and the design of the model.

A good starting point is to determine the purpose of the model. The L1DF tracker software model was to be used to evaluate and compare the performance of various data fusion algorithms in order to support the selection of the best candidates for the subsequent development of a real time prototype. Consequently, one of the main objectives pursued when developing this software model was *openness*. This means that the model has been developed to facilitate the integration of new components (such as new algorithms). Since it wasn't a requirement to run the L1DF application as a real time system, the computational speed of the application has not been a main objective.

Past experience with L1DF models has shown that testing a new data fusion algorithm is easier if it is first tested in a separate module before its integration in the overall L1DF system. Hence, the *modularity* of the model was another objective pursued. Even after the

14

integration of a tested module, it is still very useful to be able to extract it easily, thereby increasing the module *reusability*, for further testing or any other reasons.

2.2.1   Generic L1DF System

To provide a basis for the development of well-adapted, highly flexible computer-based tools to support R&D in the field of L1DF, a generic system for L1DF has been introduced that properly fuses the available source data into a tactical picture, exploiting the available a priori knowledge, while it is independent as much as possible of the specific source suite and knowledge being used. The system is generic because it identifies the main set of functions that together generally achieve L1DF, not a set of specific algorithms or techniques that could be selected and used to implement the functions. Figure 8 shows this generic L1DF system.
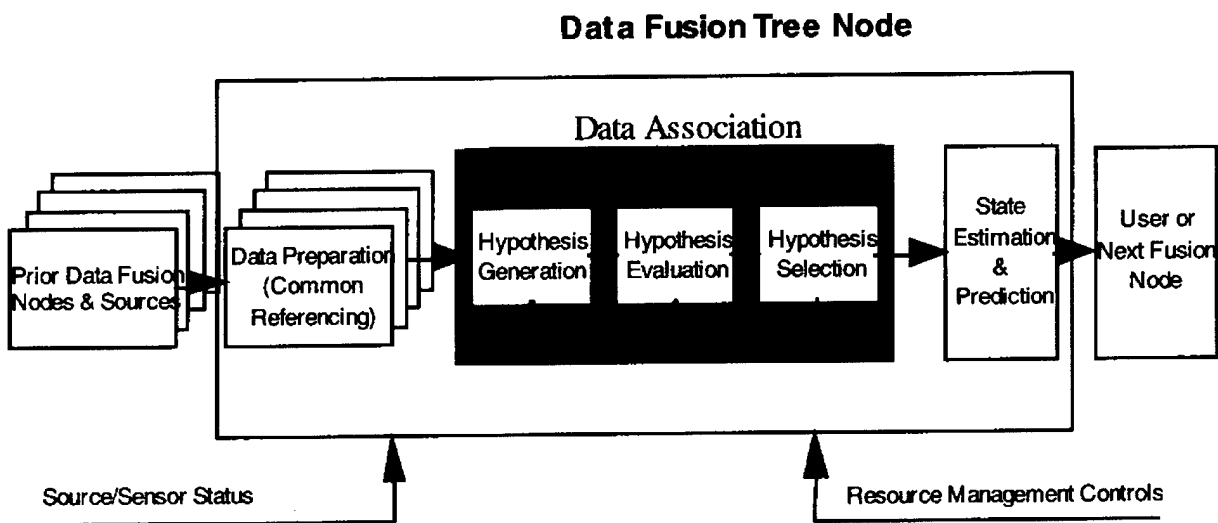
Figure 8. Generic L1DF system

In general, the task involved in a L1DF system can be classified into three categories: data preparation, data association, and state estimation.

- Data preparation: the detail tasks include

  Time alignment

  Space alignment

  Coordinate system alignment

- Data association: the detail tasks include

  Hypothesis generation (HG)

  Data gating

  Hypothesis evaluation (HE)

  Hypothesis selection (*HS*)

- State estimation

  Kinematics estimation

  Identity estimation

2.2.2 Structure of CASE_ATTI L1DF Module

In CASE_ATTI, we usually refer to a L1DF system as a "tracker". Figure 9 shows the high-level view of the tracker class structure. The main components of the OO model are the:

- *Tracker* class,
- *Input Preparation Manager* class,
- *Potential Assignation Manager* class,
- *System Track Handler* class,
- *Assignation Manager* class, and the
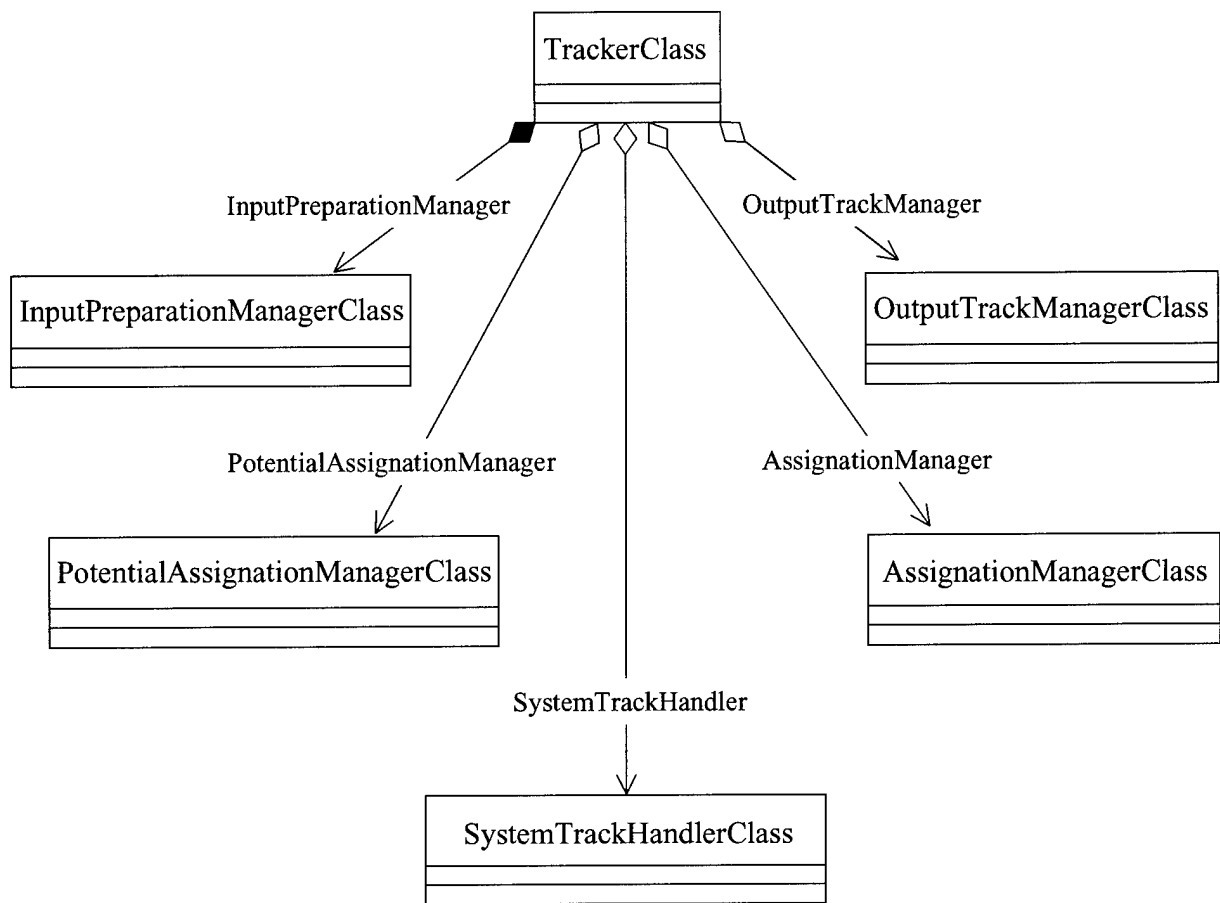- *Output Track Manager* class.

Figure 9. L1DF tracker structure

The *Tracker* has the responsibility to coordinate the software calls between the other main components. For instance, there is no direct interaction between the *System Track Handler* and the *Assignation Manager*. Indeed, the *System Track Handler* and the *Assignation Manager* are totally independent and don't have any knowledge of each other. If an interaction is needed, then the *Tracker* will get the information from one component and send it to the other components. The main advantage of applying this design strategy is to help maintain the model modularity. One main drawback is to reduce the execution speed of the application execution since a larger number of indirect calls are needed.

Figure 9 shows that the *TrackerClass* has five child classes:

- *InputPreparationManagerClass*,

- *PotentialAssignationManagerClass*,

- *SystemTrackHandlerClass*,

- *AssignationManagerClass*, and the

- *OutputTrackManagerClass*.

The *Input Preparation Manager* is related to the *Input Data Preparation* process from the high-level functional decomposition. The *Input Preparation Manager* has the responsibility to implement the various interface and buffering mechanisms, to implement the grouping tasks to create input data sets for the subsequent processing functions, and to provide a source input data activity control capability. However, based on the current functional decomposition, it does not have the responsibility to space or time align the input reports. The various data alignment activities are instead performed within specialized modules that are contained in the *Potential Assignation Manager*, the *System Track Handler* and the *Assignation Manager*.

The combined effect of the *Potential Assignation Manager* with the *Assignation Manager* is equivalent to that of the *Data Association* process from the high-level functional decomposition. The *Potential Assignation Manager* is responsible to produce a list of potential assignations between the newly received input data elements and the current system tracks. From these potential assignations, the *Assignation Manager* is responsible for selecting the proper definitive assignations that will be used within the fusion algorithms for track updating. The *Assignation Manager* may also have the responsibility to partition the entire set of system tracks and input data elements into separate clusters. This is referred to as *Cluster Management*.

The *System Track Handler* has the responsibility to manage the *System Track Selection* and the *Data Fusion* (kinematics and identity information fusion), and to perform *Track Management*.

2.2.3 Mapping from Generic L1DF System to CASE_ATTI L1DF Structure

Table 1 shows the mapping relationship from a generic L1DF system to the CASE_ATTI L1DF structure. The major L1DF tasks introduced in section 3.1 are listed in the first column and the corresponding CASE_ATTI L1DF classes are placed in the second column. Based on this table, users know where to find the proper parameters in CASE_ATTI when they need to change some characteristics of their L1DF system or tracker.

Table 1. Mapping of General L1DF system and CASE_ATTI L1DF module

General L1DF system		CASE_ATTI L1DF module
Input data preparation		(1) Input Preparation Manager Class
Data alignment		(2) Potential Assignation Manager Class (3) System Track Handler Class
Data Association	H_G	(2) Potential Assignation Manager Class
	Data gating	(2) Potential Assignation Manager Class
	H_E	(4) Assignation Manager Class
	H_S	(4) Assignation Manager Class
State estimation		(3) System Track Handler Class

2.2.5 User-interface of CASE_ATTI L1DF Module (Tracker Editor)

File

- ⊙ InputPreparationManagerClass
- ⊙ PotentialAssignationManagerClass
- ⊙ SystemTrackHandlerClass
  - ⊙ SystemTrackHandlerIOClass
    - ⊙ SystemTrackHandlerClass
      - SystemTrackHandlerInitClass
      - ⊙ FilterManagerClass
        - FilterManagerInitClass
        - ⊙ FilterClass
          - ⊙ FilterIOClass
    - ⊙ TimeAlignerParameterClass
    - ⊙ SystemTrackSectorManagerClass
    - ⊙ SystemTrackStatusEvaluatorClass
    - ⊙ SystemTrackQualityManagerClass
    - ⊙ SystemTrackObservationAttemptM
    - ⊙ IdentityCombinerManagerClass
      - ⊙ IdentityCombinerIOClass
        - ⊙ BasicDempsterShaferCombir
- ⊙ AssignationManagerClass
  - ⊙ AssignationManagerIOClass
    - ⊙ MHBasicAssignationManagerClass
      - ⊙ MHBasicAssignationManagerInitCla
      - ⊙ TimeAlignerParameterClass
      - ⊙ LikelyhoodEstimatorClass
      - ⊙ MHInputDataManagerClass
      - ⊙ AffectationManagerClass
      - ⊙ HypothesisManagerClass
      - ⊙ InterpretationClass

Tracker.txt loaded.

⇧ SELECT: FilterIOClass   ⇩ 8 Children

FilterType | IMM_FILTER_TYPE

STANDARD_KALMAN_FILTER_TYPE
PAR_MANEUVER_STANDARD_KALMAN_FILTER_TYPE
CONVERTED_MEASUREMENT_KALMAN_FILTER_TYPE
PAR_MANEUVER_CONV_MEAS_KALMAN_FILTER_TYPE
ALPHA_BETA_FILTER_TYPE
IMM_FILTER_TYPE
GPB1_MM_FILTER_TYPE
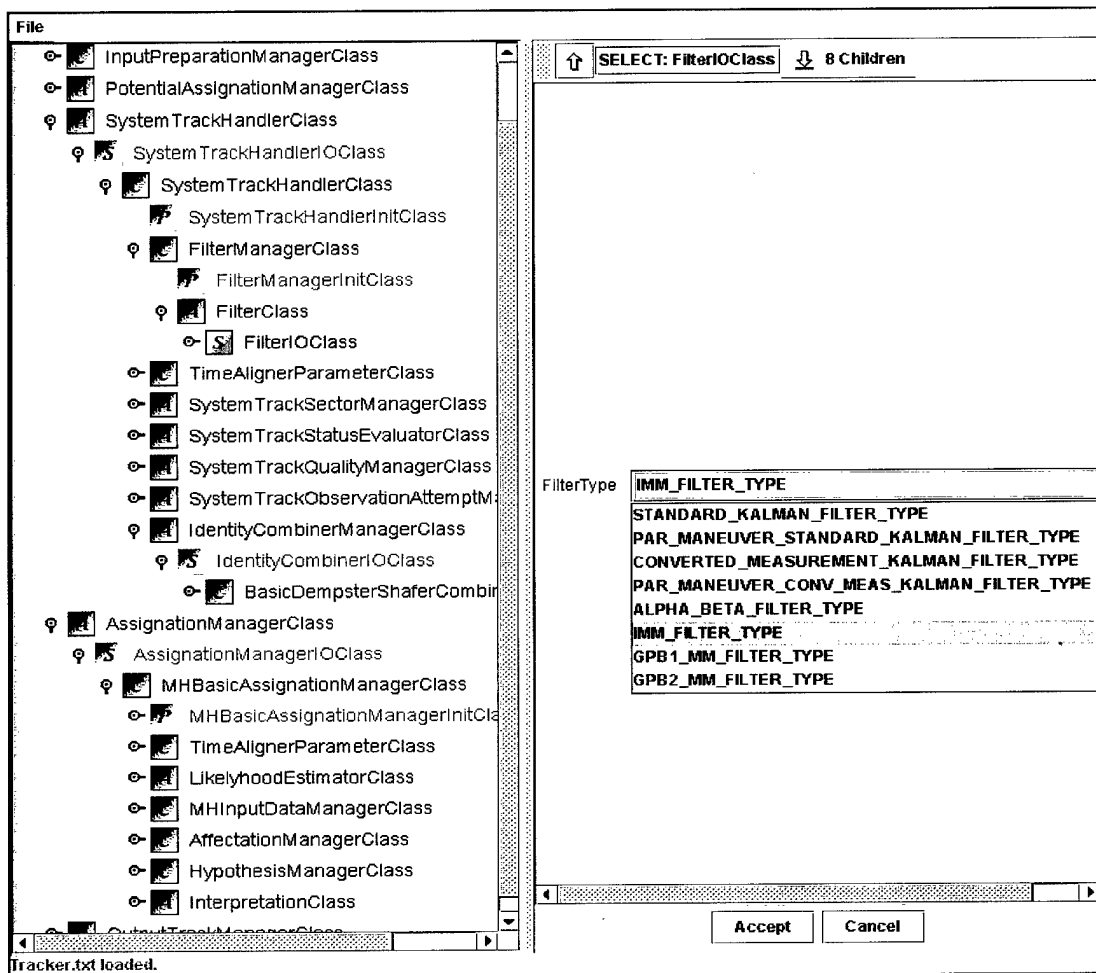GPB2_MM_FILTER_TYPE

Accept   **Cancel**

Figure 10. User-interface of CASE_ATTI L1DF Module

(Selection of the Filter Type)

2.2.6 Connection Scenario Editor

Using this module, users can select which sensor of the stimulation module is connected to which tracker of the L1DF module. These connections define the "sensor" portion of the "fusion tree". The output of each sensor is directed to the input of the L1DF module automatically. In addition, users can connect sensors to trackers in order to process their

21

data. Users then connect the trackers to the output in order to evaluate their performance with the PE module.
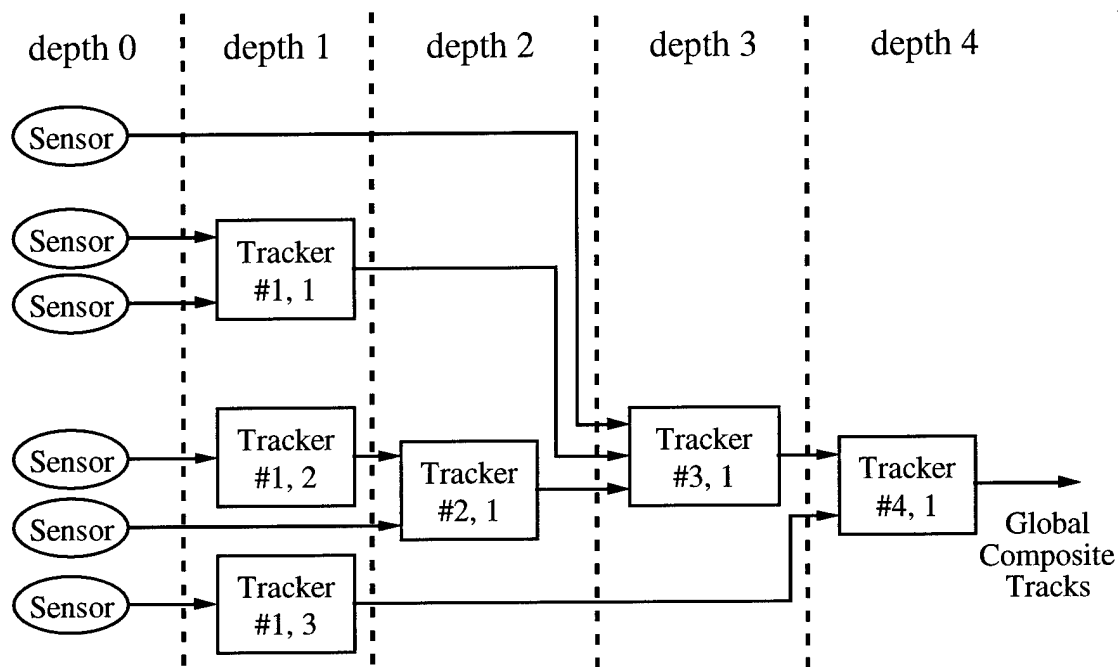

Figure 11. A complex L1DF architecture

In CASE_ATTI, the *tracker* is contained within a higher level of abstraction, the *L1DF module*, that is responsible for the reception of the buffers from the sources (i.e., from the *stimulation module*) and to redirect these buffers to the proper *tracker*. This strategy is necessary in the context of a complex L1DF architecture where one or many sources can be connected to one or many trackers, as in Figure 11.

Using this strategy, a simpler track-level fusion architecture with two autonomous sensors would be investigated in CASE_ATTI as illustrated in Figure 12. It is composite of two "sensors" (in the stimulation module) and three "trackers" (in the L1DF module).

22
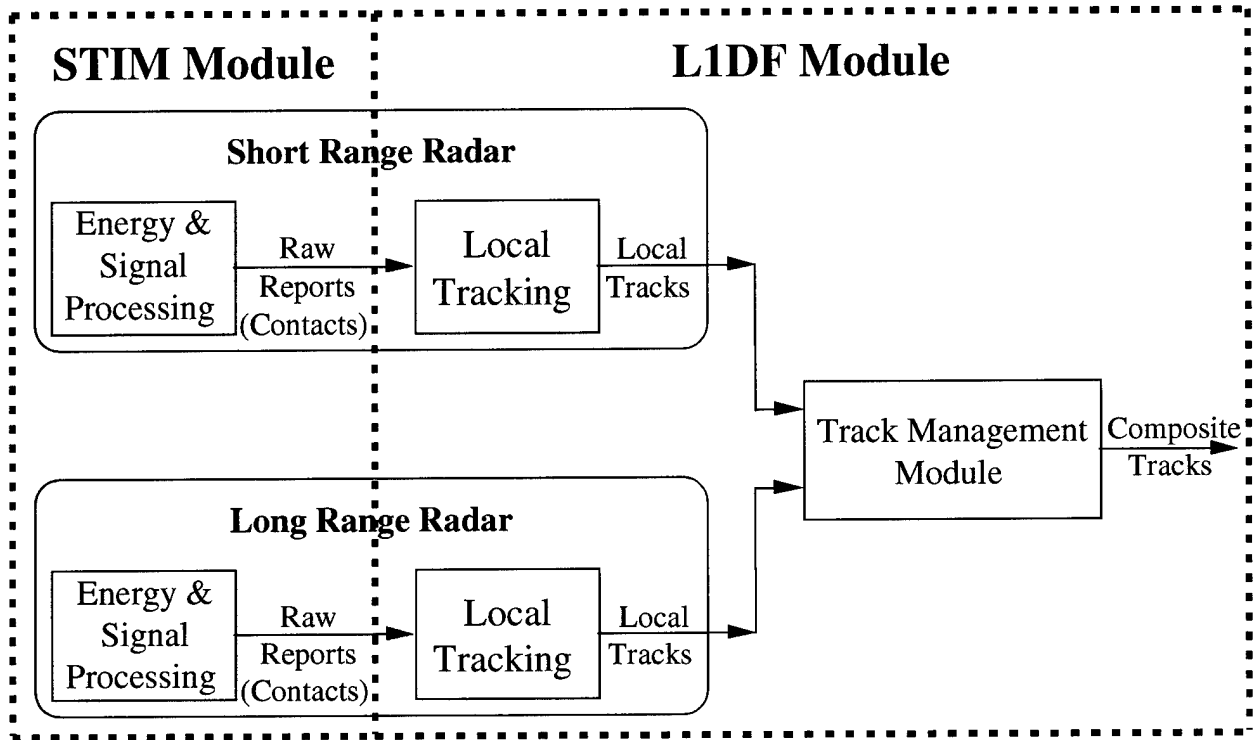
Figure 12. Track-level fusion in CASE_ATTI

However, the L1DF architecture is more complex if it is required to process the contact data from Sensor 1 in a given L1DF tracker, the contact data from Sensor 2 in a different L1DF tracker, and then merge the track data from these two trackers in a third one (as in Figure 12). This type of structure reflects what is usually called a "Track Fusion" approach (fusion of estimates rather than sensor contact data). Thus, a higher-level abstraction such as the L1DF module, enclosing a lower-level abstraction such as the tracker, enables the modeling of complex L1DF architectures.

2.2.7 User-interface of Connection Scenario Editor

Figure 13 shows an example of how users can define the data flow from "sensors" to "trackers" and finally the "output". There are two platforms in this example and each platform has two sensors on board. In the tracker structure, there are two levels of

trackers. The two bowman_contact_fusion_2 are level-one trackers and the bowman_track_fusion is level-two tracker. Those gray arrows represent the data flow that has been predefined. Users can change the undetermined data flow by moving those green and yellow arrows.
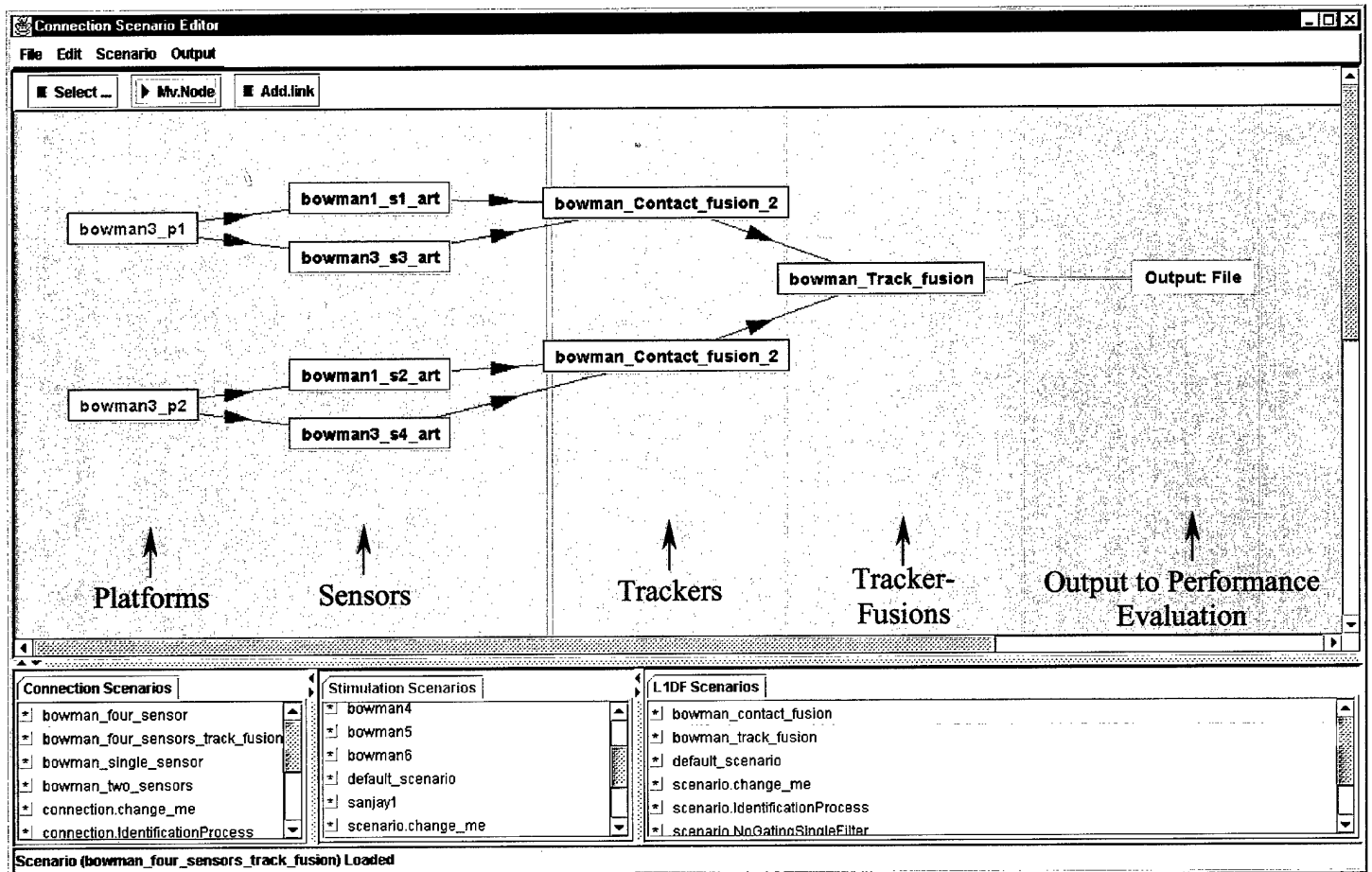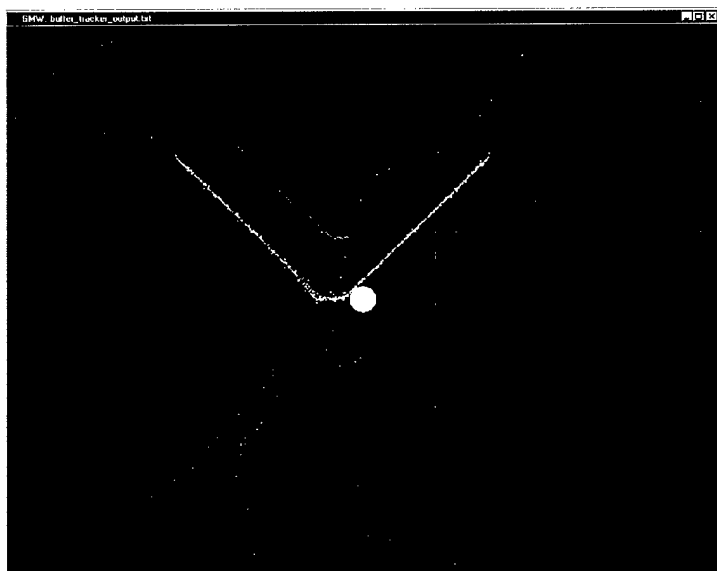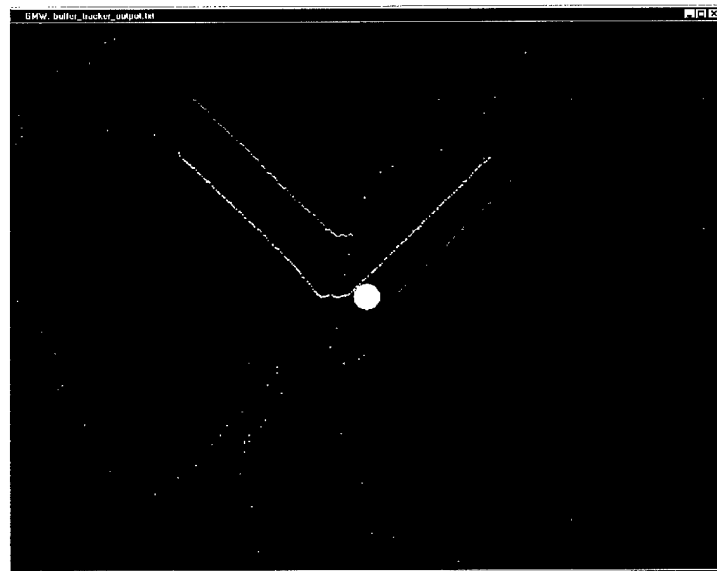

Figure 13. The user-interface of the connection scenario editor
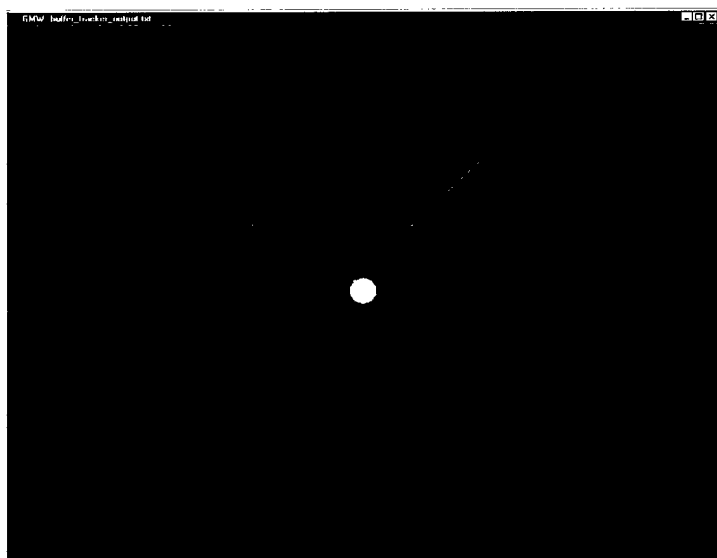
2.2.8 Results of the L1DF Module

Figure 14 shows the graphic results from the CASE_ATTI L1DF module. At this point, users can also see the outcomes of estimations (or called tracks) generated by trackers (Figure 14 (b)).
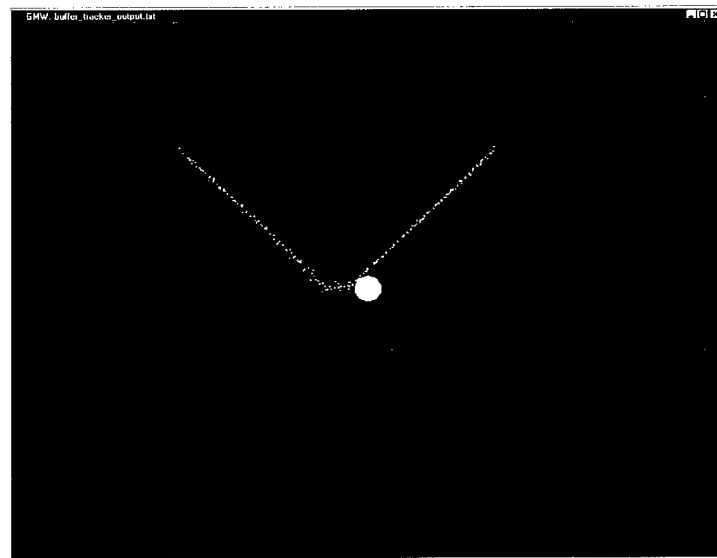
24

(a) True-tracks + contacts + Estimated tracks

(b) Estimated-tracks

(c) True-tracks

(d) Contacts

Figure 14. Graphic viewer of CASE_ATTI (L1DF Module)

2.3 Performance Evaluation (PE) Module

2.3.1 Performance Analysis Database

As shown in Figure 2, a Performance Analysis Database (PADB) retains archives of all the data manipulated by the CASE_ATTI test bed. The archived data for a given test run include the data produced by the L1DF module and the stimulation data. The PADB can also archive performance evaluation (PE) data from the PE module.

Ultimately, the performance of L1DF algorithms is judged by the success (or lack thereof) of the mission they support. However, such a global performance assessment is not appropriate during L1DF system development. With a complex L1DF system comprised of many algorithms and processes, the system designer needs specific tests to untangle the effectiveness of any individual component.

A Performance Evaluation (PE) module thus provides a set of tools in CASE_ATTI to assist the L1DF designer in his quantitative assessment of the performance of the L1DF system as a target tracking and identification process (i.e., does the L1DF system compute the "correct", "expected" output given the stimulation data provided?). These tools comprise a compilation mechanism for the tracking statistics, computation mechanisms for the various measures of performance (MOPs), etc. Figure 15 is the user interface of the PE module provided by CASE-ATTI. The background shows the list of executed scenarios (or test runs) saved in database. After users press the right mouse button on any of these executed scenarios, the menu of the 5 available MOPs will pop up.

2.3.2 Measures of Performance (MOPs)

CASE_ATTI provides 5 types of MOPs allowing users to evaluate the performance of a L1DF system (tracker). These include:

- Radial Miss Distance (RMD)
- Accuracy Filter Calculated Covariance (AFCC)
- State Estimation Error (SEE)

26

- Track Purity (TP)

- Identity

Figure 15. User-interface to the PE module

2.3.2.1 Radial Miss Distance (RMD)

At a given time, if $\mathbf{x}_t$ is the estimated position of the track t, and $\mathbf{x}_g$ the actual position of ground truth target g, then the radial miss distance is just the Pythagorean distance between them:

$$\text{RMD}(t,g) = \left| \mathbf{x}_t - \mathbf{x}_g \right|$$

27

2.3.2.2 Accuracy Filter Calculated Covariance (AFCC)

The general idea behind this MOP is to compare the covariance matrices between different variants of the MSDF system. Each sensor measurement or estimated parameter has associated uncertainties. These uncertainties can be represented for a track as ellipses (i.e., the covariance matrix are converted into error ellipses for a given confidence level) in the X-Y space for positional data and in speed-bearing space for velocity data. The accuracy of the tracks can be evaluated considering the area of those ellipses and comparison can be made between MSDF systems. This comparison will provide an approximate evaluation of the relative performance of the tracking algorithms.

2.3.2.3 State Estimation Error (SEE)

Given the assignment of tracks to truth for various evaluation times, the accuracy of the track estimate can be evaluated. For each of the track-target pairs that are associated, one computes the state estimation error as the difference between the true target state x (k) and the track estimated state at time k given measurements up to time n:

$$\tilde{x}(k \mid n) = x(k) - \hat{x}(k \mid n)$$

The estimation error is computed for each component of the state vector including velocity and acceleration as opposed to RMD that is only a positional MOP.

2.3.2.4 Track Purity (TP)

Track purity is a concept that assesses the percentage of correctly associated measurements in a given track, and so evaluates the association/tracking performance. This MOP is not explicitly dependent on detection performance, but it is dependent on the setting of association gates (and thus a function of the average innovation standard deviation which depends on P_d). It is also dependent on the target density. Track purity determines the consistency with which a track is updated with measurements from a single target or a set of targets.

2.3.2.5 Identity

The purpose of a classification MOP is to score how well an MSDF algorithm is identifying targets. It is assumed that the MSDF algorithm whose performance is to be evaluated has an identification-classification capability. It is also assumed that this classification system is based on a Dempster-Shafer type of representation scheme.

CHAPTER 3 TEST EXAMPLE

3.1 Create the Scenario

Here we created a scenario used to demonstrate how the PE module operates in CASE_ATTI. There are two targets (airplanes) flying straight and crossing to each other as shown in Figure 16. Table 2 shows the parameter settings to the stimulation module and L1DF module of the test scenario.

Table 2. Parameter settings of the test scenario

		Scan RPM (cycles/min)	Probability of detection	False alarms	Trajectory
Platform 1 (Stationary groundradar)	Sensor 1	60	0.8	NO	
	Sensor 2	40	0.8	NO	
Platform 2 (Stationary groundradar)	Sensor 3	45	0.9	NO	
	Sensor 4	55	0.9	NO	
Target 1 (Airplane)					Crossing
Target 2 (Airplane)					Crossing

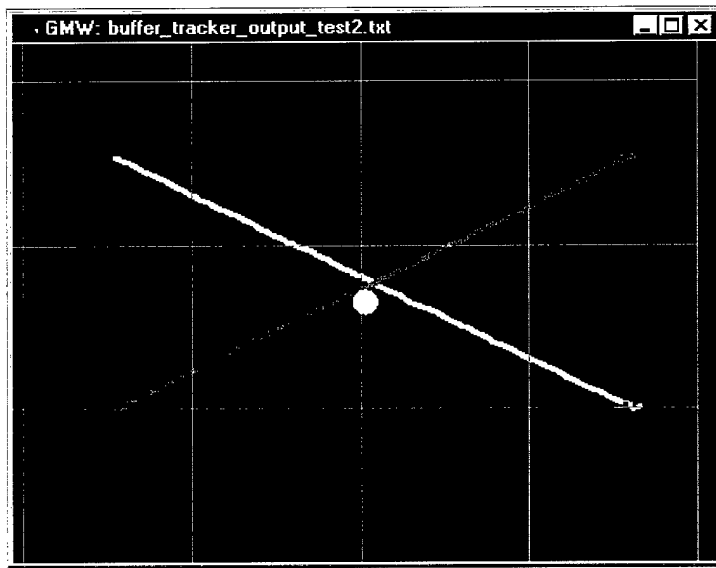|	Type of filter	Type of gate	Scoring method
Tracker	Standard	Ellipsoidal	Statistical-distance
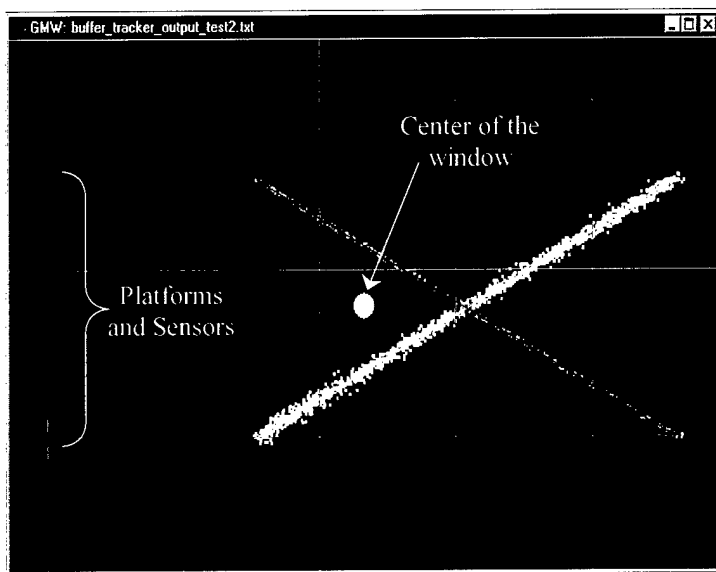
(a) True-tracks + contacts + estimated-tracks
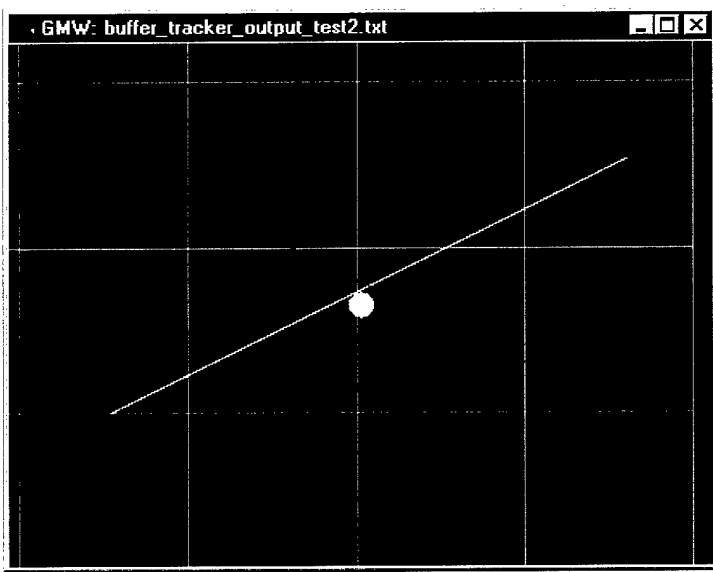
(b) True-tracks
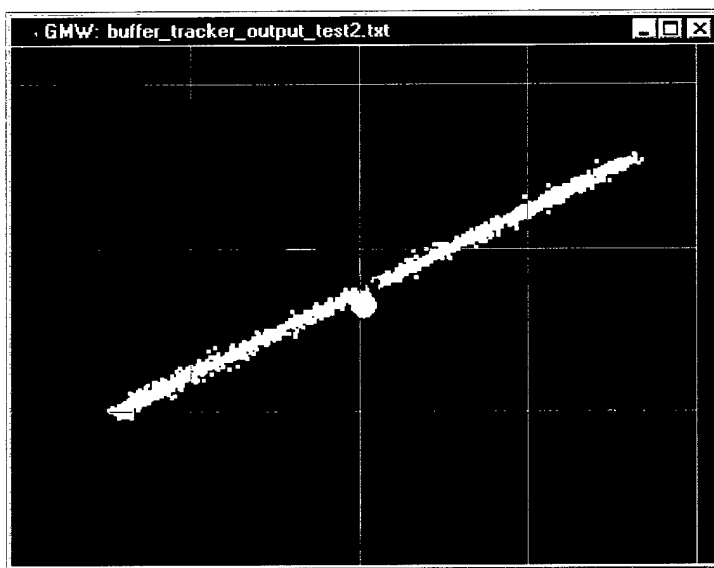
(c) Contacts

(d) Estimated-tracks

Figure 16. Graphic viewer of CASE_ATTI (test scenario)

31

3.2 Evaluate the Scenario

As shown in figure 17, there are two true-tracks (truth 1 and truth 2 in the left window) and two estimated-tracks (track 1 and track 2 in the right window). In this simple case, we can be sure that track 1 is corresponding to truth1 and track 2 is corresponding to truth 2. However it may not always be the case. Usually, users do not know the correspondence between truth and tracks; this problem of track-to-truth assignment has been a significant area of research for our project and will be discussed in a separate report.
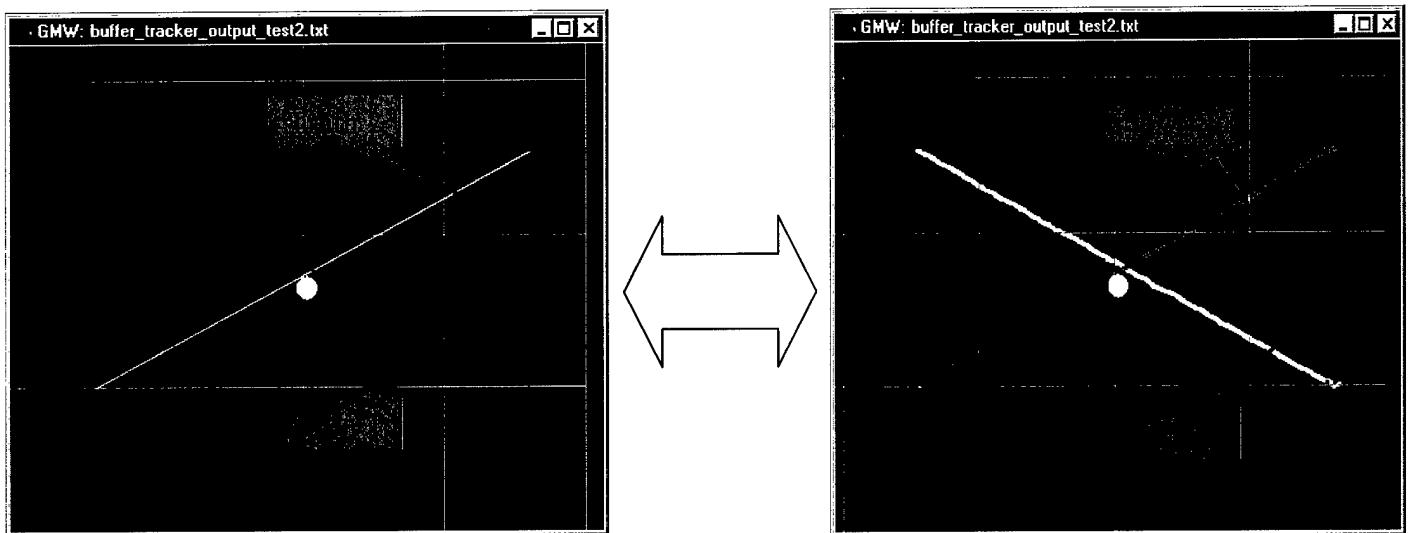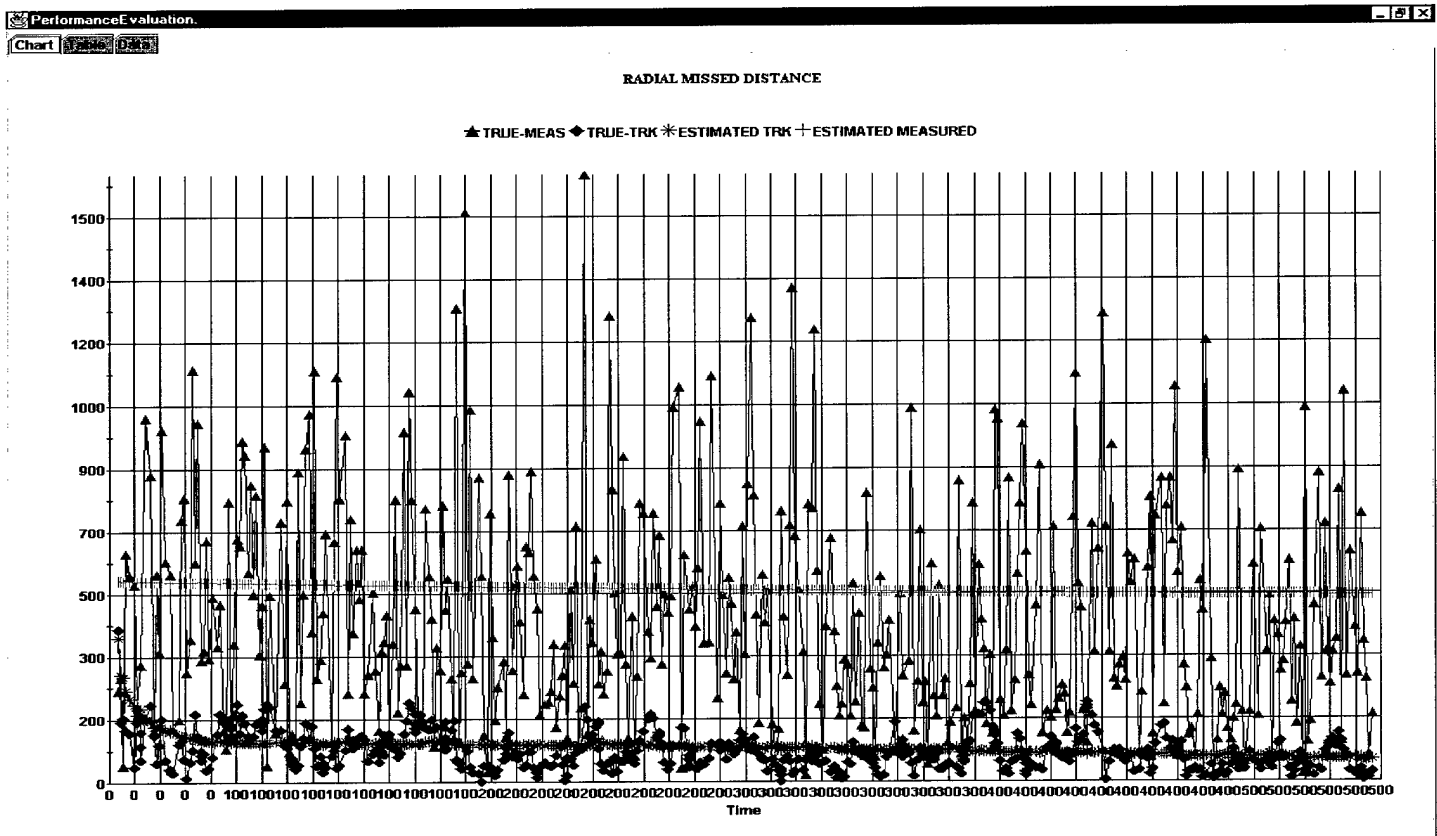

Figure 17. Truth – track assignment to the test scenario

To determine which track is corresponding to truth 1, we need to compare the differences for every track to the truth 1. In this example, they are (track 1 – truth 1) and (track 2 – truth 1). Then the radial miss distance (RMD) of these two assignments is shown in Figure 18 and Figure 19.

RADIAL MISSED DISTANCE

▲ TRUE-MEAS ◆ TRUE-TRK ✳ ESTIMATED TRK ＋ ESTIMATED MEASURED

Id	Serie	0	1	2	3
-1		TRUE-MEAS	TRUE-TRK	ESTIMATED TRK	ESTIMATED MEASURED
0	MIN	18.9584804	3.7054014600000005	56.72999207	504.69144200000005
1	MAX	1642.54802899	415.5395143	391.15099386	546.53311238
2	MEAN	445.49172543000003	84.24763631000002	94.72502065	521.88180682
3	STD	286.90321857	50.88024807	28.5851644	12.01738832

Figure 18. The radial miss distance of the track1-truth1 assignment

33

RADIAL MISSED DISTANCE

—TRUE-MEAS ┼ TRUE-TRK ⊖ ESTIMATED TRK ⊟ ESTIMATED MEASURED

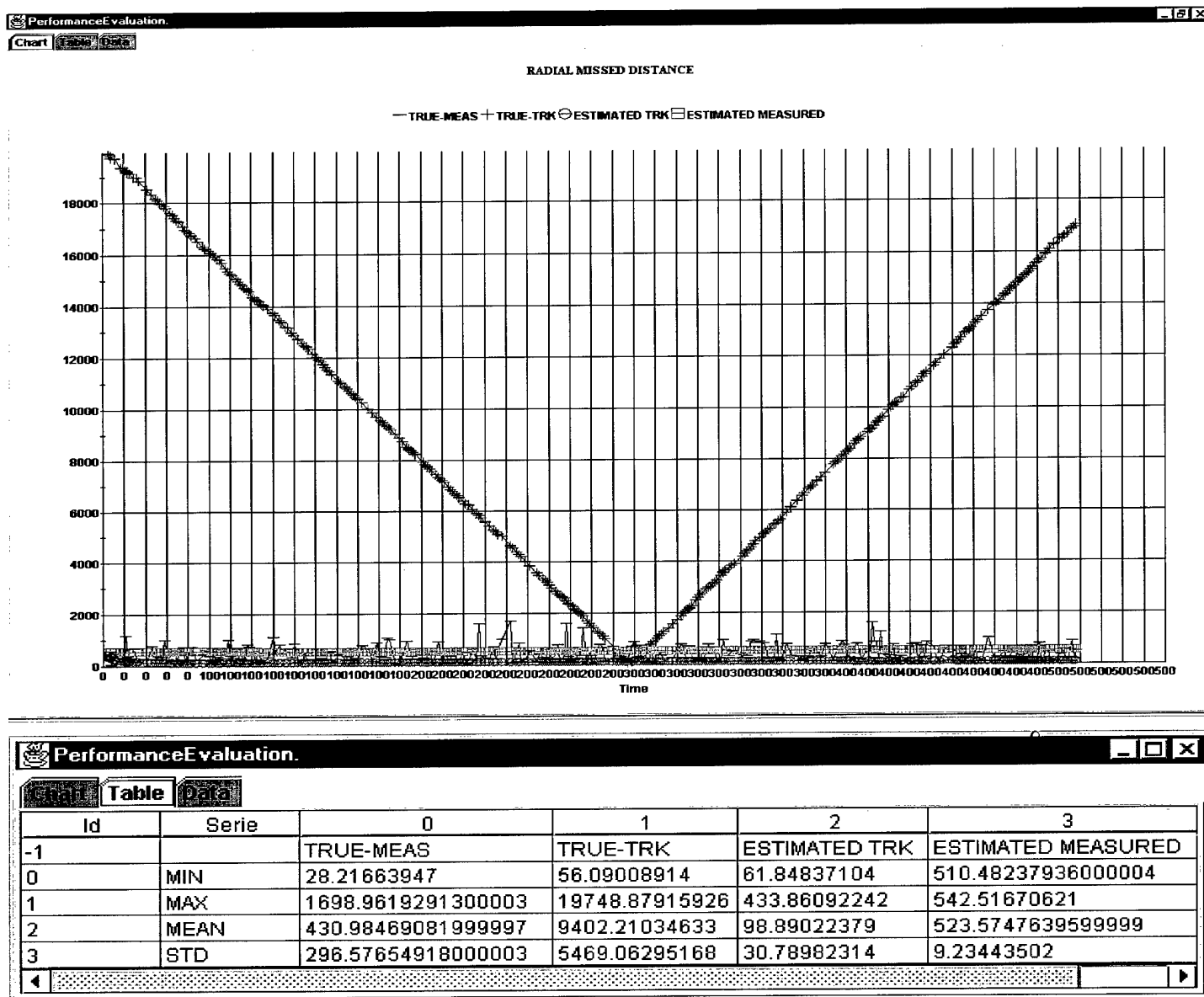| Id	Serie	0	1	2	3
-1		TRUE-MEAS	TRUE-TRK	ESTIMATED TRK	ESTIMATED MEASURED
0	MIN	28.21663947	56.09008914	61.84837104	510.48237936000004
1	MAX	1698.9619291300003	19748.87915926	433.86092242	542.51670621
2	MEAN	430.98469081999997	9402.21034633	98.89022379	523.5747639599999
3	STD	296.57654918000003	5469.06295168	30.78982314	9.23443502

Figure 19. The radial miss distance of the track2-truth1 assignment

34

Finally, after comparing the RMD in Figure 18 and Figure 19, users can determine that track 1 is corresponding to truth 1 because it has much smaller error (the mean track-to-truth distance is 84.25 meters) than the track2 – truth1 assignment (the mean track-to-truth distance is 9402.21 meters).

CHAPTER 4 CONCLUSION

As mentioned in section 4.1, CASE_ATTI provides quantitative comparisons (5 MOPs) to evaluate how tracks match truths; however, the truth-track assignment task has to be done by users manually. In other words, CASE_ATTI will not tell users which track should correspond to which truth based on its own automatic calculation. When the test scenario is simple (like the scenario shown in previous section), it is probably not a problem to users. Yet, when the scenario is too complicated, users will not be able to handle the truth-track assignment task. This is a major drawback of CASE_ATTI.

References

1. Waltz, E. and Llinas, J., *Multi-Sensor Data Fusion*, Artech House Publishes, Boston, MA, 1990

2. Roy, J., Bossé, É. and Dion, D., *PowerPoint File for CASE_ATTI Introduction*, DREV, Jan. 2000

3. Blackman, S., *Multiple Target Tracking with Radar Application*, Artech House Publishes, Boston, MA, 1986

4. Balckman, S. and Popoli, R., *Design and Analysis of Modern Tracking Systems*, Artech House Publishes, Boston, MA, 1999